AMENDMENTS TO THE CLAIMS

CLAIM AMENDMENTS

This listing of claims replaces all prior versions, and listings, of claims in the application.

1. (Currently amended) A control processor for executing a set of control tasks defining dynamic model-based interactive control of an industrial process, the control processor comprising:

an embedded control task~~, performed at a relatively low execution priority status within the control processor, the embedded control task~~ comprising a multivariable linear program including a set of output values corresponding to process setpoints; and

a set of control blocks~~, performed at a relatively high execution priority status within the control processor, the set of control blocks~~ including regulatory control blocks having output values that are transmitted by the control processor to field devices coupled to the industrial process, wherein the embedded control task executes at a lower execution priority than an execution priority of the set of control blocks.

2. (Original) The control processor of claim 1 wherein the set of control blocks comprise supervisory control blocks.

3. (Original) The control processor of claim 2 wherein the supervisory control blocks include a multivariable control block including computer instructions facilitating communication of data between the control processor and a workstation.

4. (Original) The control processor of claim 3 wherein the multivariable control block includes computer instructions for receiving and storing a process control model to be implemented by the embedded control task.

5. (Original) The control processor of claim 2 wherein the supervisory control blocks include at least one multivariable loop block including computer instructions for providing an input value for a regulatory control block.

6. (Currently amended) The control processor of claim 5 wherein the regulatory control block is a ~~PID~~proportional-integral-derivative block.

7. (Original) The control processor of claim 5 wherein the regulatory control block is a ratio block.

8. (Previously presented) The control processor of claim 1 further comprising a repetition cycle parameter specifying a period for re-commencing a cycle of the embedded control task.

9. (Original) The control processor of claim 8 wherein the set of control blocks includes a supervisory control block including a sequence of instructions to determine when to re-commence a cycle of the embedded task in accordance with a value specified by the repetition cycle parameter.

10. (Original) The control processor of claim 1 further comprising a block processing cycle parameter specifying a repetition period for re-commencing a cycle of executing the set of control blocks.

11. (Original) The control processor of claim 10 further comprising a repetition cycle parameter specifying a period for re-commencing a cycle of executing the embedded control task.

12. (Original) The control processor of claim 11 wherein a period specified by the repetition cycle parameter exceeds a period specified by the block processing cycle parameter.

13. (Currently amended) A method for operating a control processor, in an industrial process control environment, to establish operating values including a set of setpoint values and a set of process control variables associated with control elements in a controlled industrial process based upon a set of input variables including process variables provided to the control processor and representing the present state of the controlled industrial process, the method comprising the steps of:

executing ~~at a lower execution priority~~, by the control processor, an embedded multivariable control application including computer instructions facilitating computing a setpoint value corresponding to a process control variable; and

executing ~~at a higher execution priority~~, by the control processor, a set of control blocks including regulatory control blocks for receiving and storing a set of process variables representing the present state of a controlled process, wherein the embedded multivariable control application executes at a lower execution priority than an execution priority of the set of control blocks.

14. (Original) The method of claim 13 wherein the set of control blocks comprise supervisory control blocks.

15. (Original) The method of claim 14 wherein the supervisory control blocks include a multivariable control block and further including the step of downloading data from a workstation to a database accessed by the multivariable control block.

16. (Original) The method of claim 15 further comprising the steps of receiving and storing, within the database accessed by the multivariable control block, a process control model to be implemented by the embedded multivariable control application.

17. (Original) The method of claim 14 wherein the supervisory control blocks include at least one multivariable loop block, and further comprising the step of providing an input value for a regulatory control block in accordance with execution of instructions and data associated with the at least one multivariable loop block.

18. (Currently amended) The method of claim 17 wherein the regulatory control block is a ~~PID~~proportional-integral-derivative block.

19. (Original) The method of claim 17 wherein the regulatory control block is a ratio block.

20. (Previously presented) The method of claim 13 further comprising the step of maintaining a repetition cycle parameter specifying a period for re-commencing a cycle of the embedded multivariable control application.

21. (Original) The method of claim 20 wherein the set of control blocks includes a supervisory control block, and further comprising the step of determining, by the supervisory control block, when to re-commence a cycle of the embedded multivariable control application in accordance with a value specified by the repetition cycle parameter.

22. (Original) The method of claim 13 further comprising the step of maintaining a block processing cycle parameter specifying a repetition period for re-commencing a cycle of executing the set of control blocks.

23. (Previously presented) The method of claim 22 further comprising the step of maintaining a repetition cycle parameter specifying a period for re-commencing a cycle of executing the embedded multivariable control application.

24. (Original) The method of claim 23 wherein a period specified by the repetition cycle parameter exceeds a period specified by the block processing cycle parameter.

25. (Previously presented) An industrial process control computer having multiple operating levels including:

a background control program execution level wherein the process control computer executes an embedded multivariable process control application, the embedded control application including instructions for executing a multivariable linear program to generate a set of values corresponding to process control variable setpoints; and

a foreground control block execution level wherein the process control computer executes a set of control blocks, at a higher execution priority level than the background control program execution level, the set of control blocks including program instructions that, when executed, receive and store a set of process variable values representing the state of a controlled process.

26. (Previously presented) A multi-level multivariable industrial process control program execution framework for an industrial control processor including:

a first cyclically executed sequence of instructions, repeatedly executed according to a first configurable repetition period and at a first level of execution priority, the first cyclically executed sequence of instructions including at least a set of instructions for calculating a setpoint value for a process control variable; and

a second cyclically executed sequence of instructions, repeatedly executed according to a second repetition period and at a second level of execution priority, the second level of execution priority exceeding the first level of execution priority, and thus enabling the control processor to temporarily suspend execution of the first cyclically executed sequence of instructions in order to execute the second cyclically executed sequence of instructions.